

Amendments to the Specification:

Please replace paragraph [0030] of the specification as published (see US 2004/0215725 A1) with the following paragraph:

[0030] FIG. 1 shows the application architecture 100 of an illustrative embodiment of a Message Inspector according to this invention. The three main components illustrated are the Websphere Application server 102, a J2EE Application Server 104 and a JMS Messaging server 106. There may be more than one JMS Messaging server, as indicated by 108. Each of Servers 102, 104, 106, and 108 can include one or more computer readable memory devices and one or more processors coupled to the memory devices as is known to those of skill in the art. Alternatively, one or more of the servers 102, 104, 106, and 108 can run on a single processor with an associated computer readable memory device as is known to those of skill in the art.

Please replace paragraph [0035] of the specification as published with the following paragraph:

[0035] The set of java beans (categories) 118, 120, 122 and 124 use RMI communication when they require interaction with JMS Messaging Server 106. Each JMS Provider's Messaging Server 106, 108 includes an installed Message Inspector Server 128. This is an activatable RMI Server (see: ~~http://java.sun.com/products/jdk/rmi~~). The Message Inspector RMI Server 128 is responsible for all interaction with the JMS Messaging Server 106 and uses the JMS API. The Message Inspector RMI Server 128 extends the "java.rmi.activation.Activatable" class, which allows it to be invoked remotely, provided it is registered with the RMI activation demon.

Please replace paragraph [0037] of the specification as published with the following paragraph:

[0037] Turning now to FIG. 2, a brief illustration of RMI processing is shown. FIG. 2 shows the RMI Runtime structure used to make a call to the method "getMessages." A top layer is the application specific layer 200, which includes the Message Inspector RMI Client 126 and the Message Inspector RMI Server 128. The next layer is the RMI layer 202 wherein the RMI Stub 204 implements the interface methods in the application client simply by relaying the method invocation 206 to the RMI Skeleton 208 on the server side. The method invocation 206 is relayed across the network layer 210 via the RMI Runtime services 212 on the Application Server 104 over a network 214 to the RMI Runtime services 216 on the Messaging Server 106. One skilled in the art will appreciate how to construct the RMI structure after reviewing this specification and the article at: http://java.sun.com/marketing/collateral/rmi_ds.html, which is incorporated herein by reference in its entirety.